

# EAGLE - CAD Schulung

## ULP und externe Programme

Seite 001: Inhaltsverzeichnis

Seite 002: ULP system() Funktion und externe Programme

Seite 003: system() Systemprogramm starten 1

Seite 004: system() Systemprogramm starten 2

Seite 005: system() Funktion und Rückgabe der Kontrolle

Seite 006: Besonderheiten

Seite 007: Ende dieser Dokumentation

Für die Erstellung des INDEX in diesem Schaltplan unbedingt die folgende Option des ULP benutzen:  
RUN e-seiten-index EAGLE-Schulung\_AZ-CAD

# Starten von System-Programmen

Um externe Programme aufzurufen, zum Beispiel die Eingabeaufforderung (DOS-BOX) vom Betriebssystem muss man in einem ULP die Funktion `system()` benutzen.

Die meisten Programme verhindern beim Aufruf mit `system()` dass man in Eagle weiterarbeiten kann, solange das Programm noch aktiv ist. Um aber auch hier die Kontrolle wieder an Eagle (ULP) zurückzugeben, muß man diese Programme über eine Batch-Datei starten. In der Batch-Datei wird die CMD.EXE mit der Option `/C` gestartet, bzw. die Batch-Datei selbst wird mit der Option `/C` gestartet. Das veranlasst das System die Batch-Datei (CMD.EXE) zu schließen um die Kontrolle wieder an Eagle (ULP) zurückzugeben.

Ein "Programm" das die Kontrolle immer an die Aufrufende Funktion zurück gibt, ist die `explorer.exe`. `system("explorer.exe")` reicht aus um ein Windows-Fenster zu öffnen und zu Eagle zurückzukehren wobei das Windows-Fenster offen bleibt.

Die Eingabeaufforderung CMD.EXE (DOS-BOX) behält ebenso die Kontrolle bis die DOS-BOX durch Eingabe von `EXIT` + Enter beendet wird.  
Beispiel : `RUN cmd-system.ulp CMD.EXE`

Vorsicht: Die CMD.EXE hat ihre Tücken.

Gibt man während der Laufzeit des Eingabeaufforderungs-Fenster (DOS-BOX) ein DOS-Kommando ein, das einen Fehler verursacht, und wird dann das Fenster mit `EXIT` + Enter beendet, wird ein Fehler-Code zurückgegeben.

Beispiel:  
im DOS-Fenster wird `QUIT`+Enter eingeben, das erzeugt einen Fehler

```
C:>quit
Der Befehl "quit" ist entweder falsch geschrieben oder
konnte nicht gefunden werden.
```

```
C:>
```

```
C:>EXIT+Enter
```

( Wird jetzt der korrekte Befehl zum Beenden der DOS-BOX eingegeben )

Damit wird das Fenster geschlossen und ein eventuell aufgetretener Fehler wird an die aufrufende Funktion `system()` zurückgegeben.

```
Eagle: cmd-system.ulp      X
Fehler 9009 wurde erzeugt von:
"CMD.EXE"

[ OK ] [ ESC ]
```

Bei einigen Programmen funktioniert die oben beschriebene Vorgehensweise, Microsoft-WORD und EXCEL dagegen brauchen eine besondere Behandlung.

Der CMD.EXE-Aufruf (DOS-BOX/Eingabeaufforderung) und automatisch beenden klappt über eine Batch-Datei in die der Programmstart mit "START Programm.exe Daten-Dateiname" geschrieben, und dann die Batchdatei selbst mit "CMD /C Laufwerk:\Pfad\Batchdatei" im ULP an die Funktion `system()` übergeben wird.

# Starten von Systemprogrammen mit Rückgabe der Kontrolle an system()

"CMD.EXE /C Programm"

Die Option /C führt dazu, dass die CMD.EXE beendet wird auch den das Programm noch läuft.

Siehe auch in der DOS-BOX unter  
HELP

.  
. .  
.

START Startet ein eigenes Fenster, um ein bestimmtes Programm oder  
einen Befehl auszuführen.

bzw. HELP CMD

```
C:>help CMD
Startet eine neue Instanz des Windows Befehlsinterpreters.

CMD [/A | /U] [/Q] [/D] [/E:ON | /E:OFF] [/F:ON | /F:OFF] [/V:ON | /V:OFF]
[ [/S] [/C | /K] Zeichenfolge]

/C      Führt den Befehl in der Zeichenfolge aus und endet dann.
/K      Führt den Befehl in der Zeichenfolge aus und endet dann nicht.
/S      Bestimmt Behandlung von Zeichenfolgen nach /C oder /K (siehe unten).
/Q      Schaltet die Befehlsanzeige aus.
/D      Deaktiviert die Ausführung von AutoRun-Befehlen von der Registrierung.
/A      Ausgabe interner Befehle in eine Pipe oder Datei im ANSI-Format.
/U      Ausgabe interner Befehle in eine Pipe oder Datei im UNICODE-Format.
/T:fg   Legt die Hinter- bzw. Vordergrundfarben fest (siehe auch COLOR ?).
/E:ON   Aktiviert Befehlserweiterungen (siehe unten).
/E:OFF  Deaktiviert Befehlserweiterungen (siehe unten).
/F:ON   Aktiviert die Ergänzung von Datei- und Verzeichnisnamen.
/F:OFF  Deaktiviert die Ergänzung von Datei- und Verzeichnisnamen.
/V:ON   Aktiviert verzögerte Erweiterung von Variablen, ! wird dabei als
Trennzeichen verwendet. Zum Beispiel wird mit /V:ON die Variable
var zur Ausführungszeit erweitert. Im Gegensatz dazu wird bei
Verwendung der Syntax var die Variable zum Zeitpunkt der Eingabe
aufgelöst (Diese Werte können z. B. in einer FOR-Schleife
unterschiedlich sein!).
/V:OFF  Deaktiviert die verzögerte Erweiterung von Variablen.
```

Hinweis: Mehrere Befehle, die durch das Befehlstrennzeichen "&&" getrennt  
sind, werden als Zeichenfolge akzeptiert, wenn diese von Anführungsstrichen  
umgeben sind. Aus Kompatibilitätsgründen hat /X die gleiche Funktion wie  
Drücken Sie eine beliebige Taste . . .

# Starten von Systemprogrammen mit Rückgabe der Kontrolle an system()

Beispiel: CMD.EXE /C START programm

Das Ausführbare Programm und der Dateiname muss in " " eingeschlossen werden, wegen Spaces im Pfad-Dateinamen  
"CMD.EXE /C START " ist nötig, da sonst auf das Beenden des Programm gewartet wird.  
sprintf(syscommand, "CMD.EXE /C START "programm" "Datei"");  
system(syscommand); // externes Programm starten

```
C:>help start
Startet ein eigenes Fenster, um ein bestimmtes Programm oder einen Befehl
auszuführen.

START ["Titel"] [/D Pfad] [/I] [/MIN] [/MAX] [/SEPARATE | /SHARED]
[/LOW | /NORMAL | /HIGH | /REALTIME] | /ABOVENORMAL | /BELOWNORMAL]
[/NODE <NUMA-Knoten>] [/AFFINITY <Hex.-Affinitätsmaske>] [/WAIT] [/B]
[Befehl/Programm]
[Parameter]

"Titel"      Der Titel des neuen Fensters.
Pfad        Startverzeichnis.
B           Startet Anwendung, ohne ein neues Fenster zu öffnen. Die
           Anwendung ignoriert STRG+C. Wenn die Anwendung nicht
           selbständig STRG+C überprüft, ist STRG+UNTBR die einzige
           Möglichkeit, um die Anwendung abzubrechen.
I           Die CMD.EXE beim Aufruf übergebene Umgebung soll die neue
           aktuelle Umgebung sein.
MIN         Startet das Fenster minimiert.
MAX         Startet das Fenster maximiert.
SEPARATE    Startet 16-Bit-Windows-Programm in separatem Speicherbereich.
SHARED      Startet 16-Bit-Windows-Programm in gemeinsamen
           Speicherbereich.
LOW         Startet Anwendung in IDLE-Prioritätsklasse.
NORMAL      Startet Anwendung in der NORMAL-Prioritätsklasse.
HIGH        Startet Anwendung in der HIGH-Prioritätsklasse.
REALTIME    Startet Anwendung in der REALTIME-Prioritätsklasse.
ABOVENORMAL Startet Anwendung in der ABOVENORMAL-Prioritätsklasse.
BELOWNORMAL Startet Anwendung in der BELOWNORMAL-Prioritätsklasse.
NODE        Legt den Knoten der bevorzugten NUMA-Architektur
           (Non-Uniform Memory Architecture) fest als Dezimalganzzahl
           fest.
AFFINITY    Legt die Prozessoraffinitätsmaske als Hexadezimalzahl fest.
           Der Prozess kann nur auf diesen Prozessoren ausgeführt werden.

           Die Affinitätsmaske wird abweichend interpretiert, wenn
           /AFFINITY und /NODE kombiniert werden. Legt die
           Affinitätsmaske so fest, als würde die Prozessormaske
           des NUMA-Knotens nach rechts verschoben, um bei Bit null zu
           beginnen. Der Prozess kann nur auf diesen gemeinsamen
           Prozessoren zwischen der festgelegten Affinitätsmaske und dem
           NUMA-Knoten ausgeführt werden. Wenn keine gemeinsamen
           Prozessoren vorhanden sind, wird der Prozess auf
           die Ausführung auf dem festgelegten NUMA-Knoten beschränkt.
WAIT        Startet die Anwendung und wartet auf ihre Beendigung.
Befehl/Programm
           Wenn es sich um einen internen "cmd"-Befehl oder eine
           Batchdatei handelt, wird der Befehlszeilenprozessor "cmd.exe"
           mit dem Parameter "/K" gestartet. Dies bedeutet, dass das
           Fenster nach der Ausführung des Befehls nicht geschlossen
           wird.

           Wenn es sich nicht um einen internen "cmd"-Befehl oder eine
           Batchdatei handelt, dann handelt es sich um ein Programm und
           wird entweder in einem Fenster ausgeführt oder als
           Konsolenanwendung.
```

# Starten von Externen Programmen mit Rückgabe der Kontrolle an system()

Beispiel: CMD.EXE /C START programm [Datei] [Optionen]

Das Ausführbare Programm und der Dateiname muss in " " eingeschlossen werden, wegen Spaces im Pfad-Dateinamen!  
"CMD.EXE /C START " ist nötig, da sonst auf das Beenden des Programm gewartet wird.  
sprintf(syscommand, "CMD.EXE /C START "programm" "Datei"");  
system(syscommand); // externes Programm starten

```
C:>help start
Startet ein eigenes Fenster, um ein bestimmtes Programm oder einen Befehl
auszuführen.

START ["Titel"] [/D Pfad] [/I] [/MIN] [/MAX] [/SEPARATE | /SHARED]
[/LOW | /NORMAL | /HIGH | /REALTIME] | /ABOVENORMAL | /BELOWNORMAL]
[/NODE <NUMA-Knoten>] [/AFFINITY <Hex.-Affinitätsmaske>] [/WAIT] [/B]
[Befehl/Programm]
[Parameter]

"Titel"      Der Titel des neuen Fensters.
Pfad        Startverzeichnis.
B           Startet Anwendung, ohne ein neues Fenster zu öffnen. Die
           Anwendung ignoriert STRG+C. Wenn die Anwendung nicht
           selbständig STRG+C überprüft, ist STRG+UNTB die einzige
           Möglichkeit, um die Anwendung abzubrechen.
I           Die CMD.EXE beim Aufruf übergebene Umgebung soll die neue
           aktuelle Umgebung sein.
MIN         Startet das Fenster minimiert.
MAX         Startet das Fenster maximiert.
SEPARATE    Startet 16-Bit-Windows-Programm in separatem Speicherbereich.
SHARED      Startet 16-Bit-Windows-Programm in gemeinsamen
           Speicherbereich.
LOW         Startet Anwendung in IDLE-Prioritätsklasse.
NORMAL      Startet Anwendung in der NORMAL-Prioritätsklasse.
HIGH        Startet Anwendung in der HIGH-Prioritätsklasse.
REALTIME    Startet Anwendung in der REALTIME-Prioritätsklasse.
ABOVENORMAL Startet Anwendung in der ABOVENORMAL-Prioritätsklasse.
BELOWNORMAL Startet Anwendung in der BELOWNORMAL-Prioritätsklasse.
NODE        Legt den Knoten der bevorzugten NUMA-Architektur
           (Non-Uniform Memory Architecture) fest als Dezimalganzzahl
           fest.
AFFINITY    Legt die Prozessoraffinitätsmaske als Hexadezimalzahl fest.
           Der Prozess kann nur auf diesen Prozessoren ausgeführt werden.

           Die Affinitätsmaske wird abweichend interpretiert, wenn
           /AFFINITY und /NODE kombiniert werden. Legt die
           Affinitätsmaske so fest, als würde die Prozessormaske
           des NUMA-Knotens nach rechts verschoben, um bei Bit null zu
           beginnen. Der Prozess kann nur auf diesen gemeinsamen
           Prozessoren zwischen der festgelegten Affinitätsmaske und dem
           NUMA-Knoten ausgeführt werden. Wenn keine gemeinsamen
           Prozessoren vorhanden sind, wird der Prozess auf
           die Ausführung auf dem festgelegten NUMA-Knoten beschränkt.
WAIT        Startet die Anwendung und wartet auf ihre Beendigung.
Befehl/Programm
           Wenn es sich um einen internen "cmd"-Befehl oder eine
           Batchdatei handelt, wird der Befehlszeilenprozessor "cmd.exe"
           mit dem Parameter "/K" gestartet. Dies bedeutet, dass das
           Fenster nach der Ausführung des Befehls nicht geschlossen
           wird.

           Wenn es sich nicht um einen internen "cmd"-Befehl oder eine
           Batchdatei handelt, dann handelt es sich um ein Programm und
           wird entweder in einem Fenster ausgeführt oder als
           Konsolenanwendung.
```

# Starten von Externen Programmen

## Besonderheiten

Das Besondere an externen (fremden) Programmen, im Gegensatz zu Systemprogrammen ist, dass sie mit dem kompletten Pfad aufgerufen werden müssen.

Da Systemprogramme eingebettet sind, wie CMD.EXE, explorer.exe ... reicht es aus einfach den Namen an das System zu übergeben, das System findet das Programm anhand der Einträge in der Registry.

Bei externen Programmen sieht es anders aus, selbst wenn sie in der Registry eingetragen sind, kann es erforderlich sein dass man den kompletten Pfad mit Namen angeben muss, auf jeden Fall aber Programme die man nicht zwingend installieren muss.

Um hier keine Fehlermeldung bezüglich nicht gefundener Programme zu erzeugen, kann man entsprechende Einträge in der eaglerc.usr vornehmen. Dazu gibt es in der User-Language (ULP) die Funktion cfgset() und cfgget().

Um nicht in Konflikt mit etwaigen Eagle-Konfigurations-Parametern zu kommen, sollte man für den Parameternamen ein "ULP:" voranstellen wie in den untenstehenden Beispielen.

Hier einige Beispiele von externen Programmen die benutzt werden können.

Auszug aus der eaglerc.usr :

```
ULP:AcrobatReader_ulp = "C:/Program Files (x86)/Adobe/Acrobat Reader DC/Reader/AcroRd32.exe"
ULP:EaglePowertoolDxfExecuteRc = "C:/Program Files (x86)/EAGLE_PCB_Power_Tools/Dxf2scr.exe"
ULP:Eagle_FreeCad_Execute_Rc = "C:/Prog/FreeCAD 0.18/bin/FreeCAD.exe"
ULP:ExecutableEditor_ulp = "C:/Program Files/Notepad++/notepad++.exe"
ULP:HPGLexecuteTerminalProgram = "D:/Daten/Hyperterm_datan/hypertrm.exe"
```

Der Aufruf dieser Programme bzw. der ULPs die diese Programme starten, sind in der eagle.scr eingetragen und definieren hauptsächlich Text-Buttons mit selbst erstellten Icons.

Die ULPs die diese Programme aufrufen sind in der Regel so gestaltet, dass sie zuerst nach dem Entsprechenden Eintrag in der eaglerc.usr suchen cfgget(), und wenn dafür keine Information zurückgelesen wird, öffnet sich ein Datei-Dialog in dem man sich zum entsprechenden ausführbaren Programm (Datei) klickt und die Datei selbst anwählt. Anschliessend wird mit cfgset() der Parameter in die eaglerc.usr geschrieben. Achtung: cfgset() schreibt nicht in die Datei selbst, sondern in ein Speicher-Array das dann beim Beenden von Eagle (ALT+X) in die eaglerc.usr geschrieben wird.

Hat man also einen entsprechenden Parameter aktuell gesetzt und dann die eagle.exe über den Taskmanager oder durch einen System-Absturz beendet, ist diese Information beim nächsten Start von Eagle nicht in der eaglerc.usr eingetragen.

Der Umgang mit diesen Parametern ist in der Hilfe beschrieben, unter:

HELP - User Language - Builtins - Builtin-Functions - Sonstige Funktionen - Konfigurations-Parameter

Weitere Informationen zur eaglerc.usr finden Sie in der Dokumentation EAGLE-Schulung.sch / EAGLE-Schulung.pdf

	1	2	3	4	5	6	7	8	9	10	11
A											
B											
C											
D											
E											
F											
G											

Vorläufiges Ende der Dokumentation 2020-10-13  
EAGLE-Schulung\_ULP\_und\_externe\_Programme  
A. Zaffran